

Scripts AWK

1. Escriba un script AWK para contar el número total de frases (*talkspurts*) en una traza de retardo.

Con el siguiente script se muestra el número total de frases para cada traza de retardo. A continuación, se muestra el total de frases para cada traza:

Traza 1- Archivo "1.txt": 57796 en total

```
awk 'END {print NR}' 1.txt
```

Traza 2- Archivo "2.txt": 24896 en total

```
awk 'END {print NR}' 2.txt
```

Traza 3- Archivo "3.txt": 38175 en total

```
awk 'END {print NR}' 3.txt
```

Traza 4- Archivo "4.txt": 28065 en total

```
awk 'END {print NR}' 4.txt
```

Traza 5- Archivo "5.txt": 53375 en total

```
awk 'END {print NR}' 5.txt
```

Traza 6- Archivo "6.txt": 23591 en total

```
awk 'END {print NR}' 6.txt
```

Utilice NR que nos indica el número de líneas que hay en el texto y solamente las imprimí y también utilice END para que solo me diera el ultimo resultado ya que si no lo pongo me iba a salir una lista larga.

2. Realice lo mismo, pero ahora para contar el número total de paquetes que llegaron al receptor.

El siguiente script muestra el número total de paquetes que llegan al receptor y también se muestra otro script en donde se muestra el número total en donde comienza un periodo de silencio.

Para el archivo – "1.txt": Hay un total de 56979 de paquetes que llegan al receptor y hay 817 periodos de silencio.

```
awk '/D/ {nlineas += 1} END {print nlineas}'
```

```
awk '/!/ {nlineas += 1} END {print nlineas}'
```

Para el archivo – “2.txt”: Hay un total de 24490 de paquetes que llegan al receptor y hay 406 periodos de silencio.

```
awk '/D/ {nlineas += 1} END {print nlineas}'
```

```
awk '/!/ {nlineas += 1} END {print nlineas}'
```

Para el archivo – “3.txt”: Hay un total de 37640 de paquetes que llegan al receptor y hay 535 periodos de silencio.

```
awk '/D/ {nlineas += 1} END {print nlineas}'
```

```
awk '/!/ {nlineas += 1} END {print nlineas}'
```

Para el archivo – “4.txt”: Hay un total de 27814 de paquetes que llegan al receptor y hay 251 periodos de silencio.

```
awk '/D/ {nlineas += 1} END {print nlineas}'
```

```
awk '/!/ {nlineas += 1} END {print nlineas}'
```

Para el archivo – “5.txt”: Hay un total de 52836 de paquetes que llegan al receptor y hay 539 periodos de silencio.

```
awk '/D/ {nlineas += 1} END {print nlineas}'
```

```
awk '/!/ {nlineas += 1} END {print nlineas}'
```

Para el archivo – “6.txt”: Hay un total de 23293 de paquetes que llegan al receptor y hay 298 periodos de silencio.

```
awk '/D/ {nlineas += 1} END {print nlineas}'
```

```
awk '/!/ {nlineas += 1} END {print nlineas}'
```

Para calcular el número de paquetes total, hice una iteración a la cual la llame “nlineas” esto va a ir checando fila por fila si contiene la letra “D” eso solo para los paquetes, pero también lo realice para el periodo de silencio “!”, igual utilice END para que solo me imprimiera el ultimo resultado.

3. Escriba un script AWK para encontrar la diferencia mínima entre la estampa de tiempo del receptor menos de la del emisor.

El siguiente script muestra la mínima diferencia de cada traza:

Archivo – “1.txt”: El valor mínimo: -641808

```
awk 'BEGIN { min1 = 0 } { resultado = $2 - $3; if(resultado < min1) min1 = resultado;} END { print min1 }' 1.txt
```

Archivo – “2.txt”: El valor mínimo: -14643186

```
awk 'BEGIN { min2 = 0 } { resultado = $2 - $3; if(resultado < min2) min2 = resultado;} END { print min2 }' 2.txt
```

Archivo – “3.txt”: El valor mínimo: -774606

```
awk 'BEGIN { min3 = 0 } { resultado = $2 - $3; if(resultado < min3) min3 = resultado;} END { print min3 }' 3.txt
```

Archivo – “4.txt”: El valor mínimo: -807010

```
awk 'BEGIN { min4 = 0 } { resultado = $2 - $3; if(resultado < min4) min4 = resultado;} END { print min4 }' 4.txt
```

Archivo – “5.txt”: El valor mínimo: 0

```
awk 'BEGIN { min5 = 0 } { resultado = $2 - $3; if(resultado < min5) min5 = resultado;} END { print min5 }' 5.txt
```

Archivo – “6.txt”: El valor mínimo: -1489551

```
awk 'BEGIN { min6 = 0 } { resultado = $2 - $3; if(resultado < min6) min6 = resultado;} END { print min6 }' 6.txt
```

Utilice BEGIN para que empezara desde t_0 declare una variable como “min” y otra llamada “resultado” en resultado se va a almacenar la resta entre la estampa de tiempo del receptor menos la del emisor, y de todas esas restas que va hacer voy a preguntarme si el resultado es menor “min” si es menor entonces “min” va a tomar ese valor de resultado y lo termine con END para que mostrara toda la lista de las restas que hizo, y después imprimí el mínimo de cada traza.

Aclaración: Declare a min igual a 0 porque todos los resultados eran negativos, si hubieran sido positivos a min lo tendría que declarar con el número total de frases de cada traza

4. Escriba un script AWK que reciba una traza como entrada y entregue un archivo con dos columnas: tiempo de la sesión en segundos, y retardo de extremo a extremo en segundos.

El siguiente script muestra el tiempo de la sesión en segundos y retardo de extremo a extremo:

Archivo - "1.txt":

```
awk ' /D/ { print ($3 - 131912079)/8000, ($2 - $3 - (-641808))/8000 }' 1.txt > data1.dat
```

Archivo - "2.txt":

```
awk ' /D/ { print ($3 - 119211478)/8000, ($2 - $3 - (-14643186))/8000 }' 2.txt > data2.dat
```

Archivo - "3.txt":

```
awk ' /D/ { print ($3 - 135895789)/8000, ($2 - $3 - (-774606))/8000 }' 3.txt > data3.dat
```

Archivo - "4.txt":

```
awk ' /D/ { print ($3 - 460510397)/8000, ($2 - $3 - (-807010))/8000 }' 4.txt > data4.dat
```

Archivo - "5.txt":

```
awk ' /D/ { print ($3 - 415181937)/8000, ($2 - $3 - (0))/8000, }' 5.txt > data5.dat
```

Archivo - "6.txt":

```
awk ' /D/ { print ($3 - 501927541)/8000, ($2 - $3 - (-1489551))/8000 }' 6.txt > data6.dat
```

En este script utilice las ecuaciones que vienen en la práctica, para eso solo indique que solo se hicieran para los paquetes recibidos "D" luego realice los cálculos de la primera columna que fue el tiempo de la sesión y después en una segundo columna realice los cálculos para el retardo de extremo a extremo, y los resultados los pase a otro archivo ya que así lo indica la práctica para después graficar esos resultados.

5. Escriba un script AWK que calcule el retardo promedio de extremo a extremo en una sesión.

Para el siguiente script utilice el archivo "data.dat" porque ya se había calculado el tiempo de sesión en el inciso anterior solo saque el promedio para cada uno de los archivos.

Archivo – "data1.dat": 675.519

```
awk ' { suma+= $1 } END {print suma/NR} ' data1.dat
```

Archivo – "data2.dat": 760.623

```
awk ' { suma+= $1 } END {print suma/NR} ' data2.dat
```

Archivo – "data3.dat": 574.162

```
awk ' { suma+= $1 } END {print suma/NR} ' data3.dat
```

Archivo – "data4.dat": 289.542

```
awk ' { suma+= $1 } END {print suma/NR} ' data4.dat
```

Archivo – "data5.dat": 545.416

```
awk ' { suma+= $1 } END {print suma/NR} ' data5.dat
```

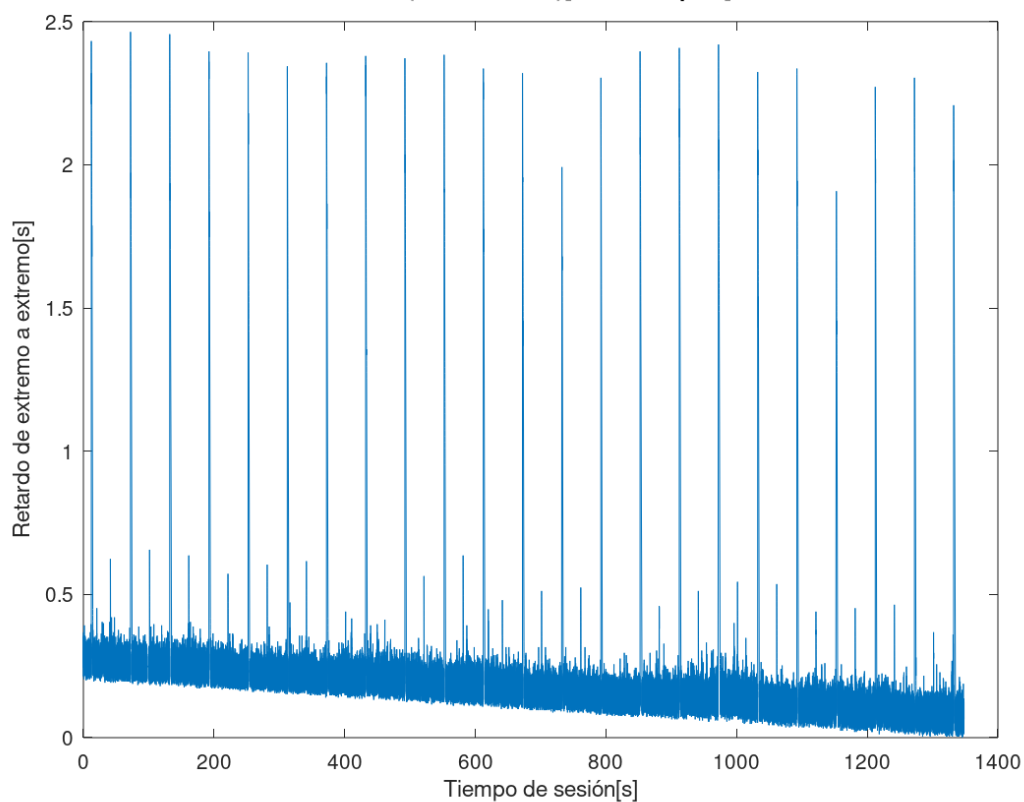
Archivo – "data6.dat": 320.781

```
awk ' { suma+= $1 } END {print suma/NR} ' data6.dat
```

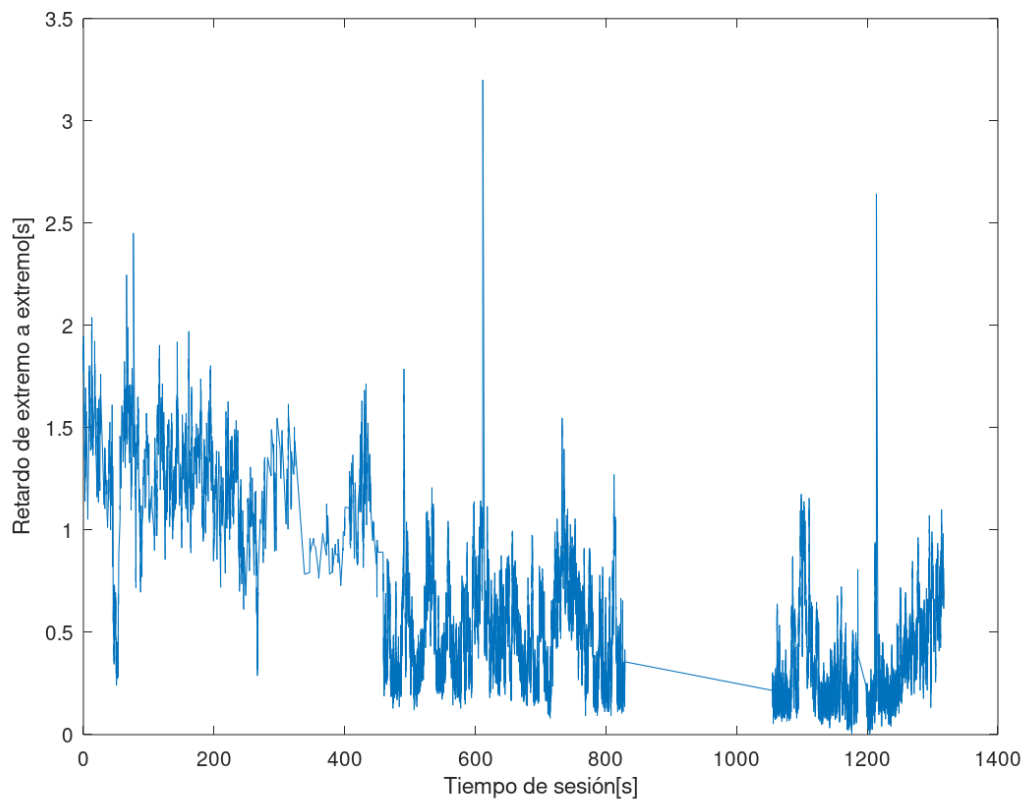
En este solo reutilice el nuevo archivo en donde se realizo los cálculo del tiempo de sesión que se encuentran en la columna 1, entonces declare una variable "suma" lo cual va a estar sumando cada fila de la columna y se va almacenar ahí, para después dividir el total de frases de cada traza (para eso volví a utilizar NR), para sacar el promedio de extremo a extremo.

Gráficas Macroscópicas

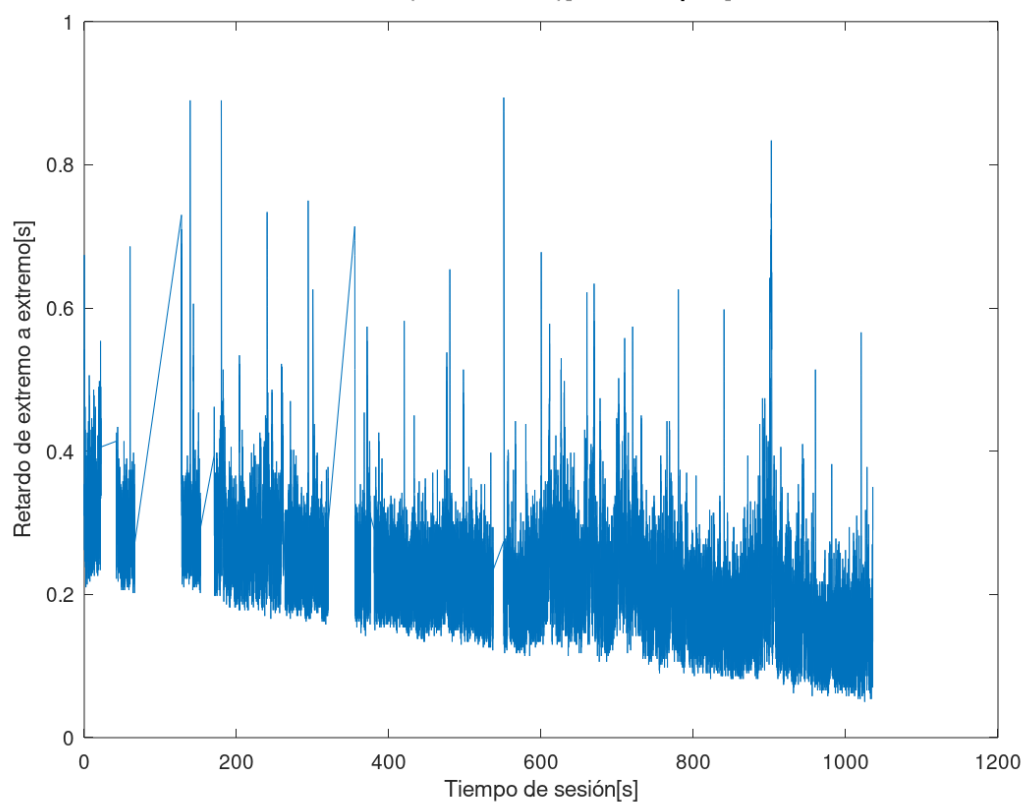
**Traza 1.- Retardo de extremo a extremo
VoIP(Voice over IP)[Macroscópica]**



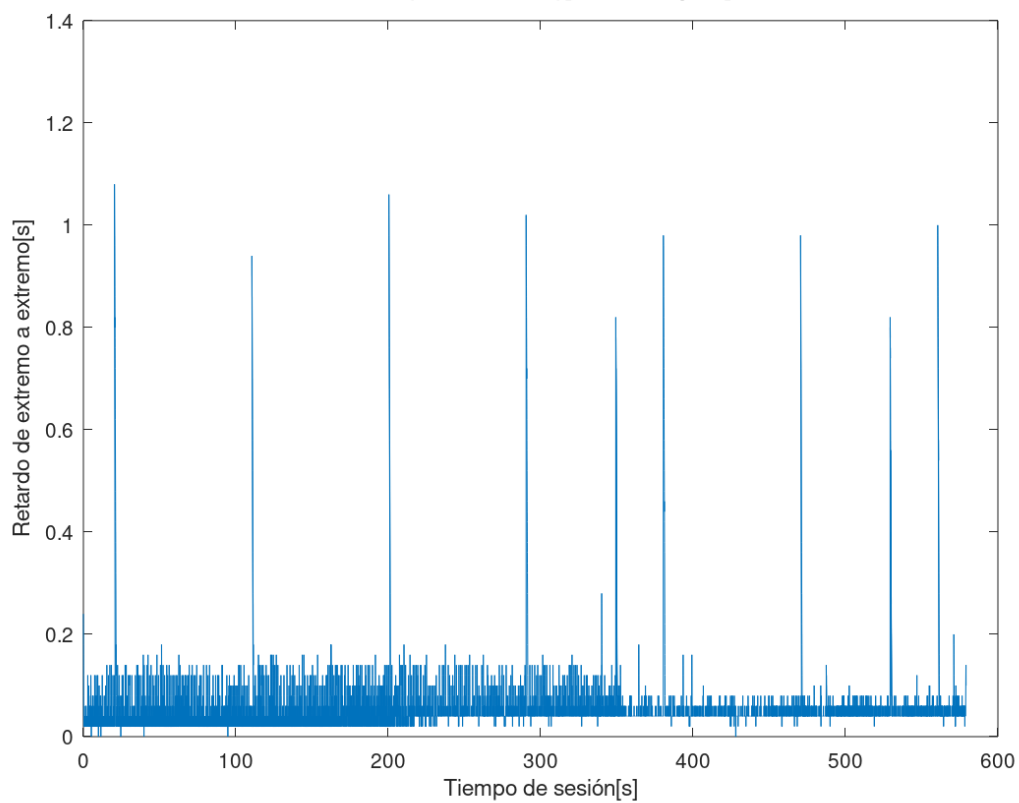
**Traza 2.- Retardo de extremo a extremo
VoIP(Voice over IP)[Macroscópica]**



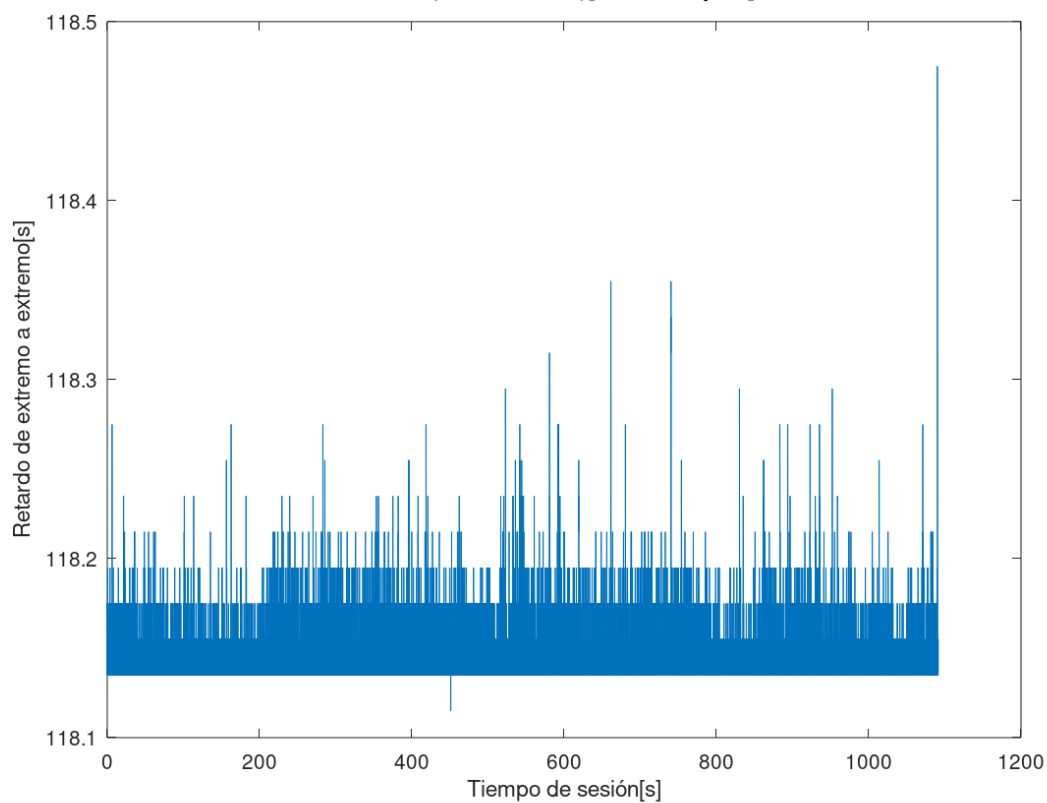
**Traza 3.- Retardo de extremo a extremo
VoIP(Voice over IP)[Macroscópica]**



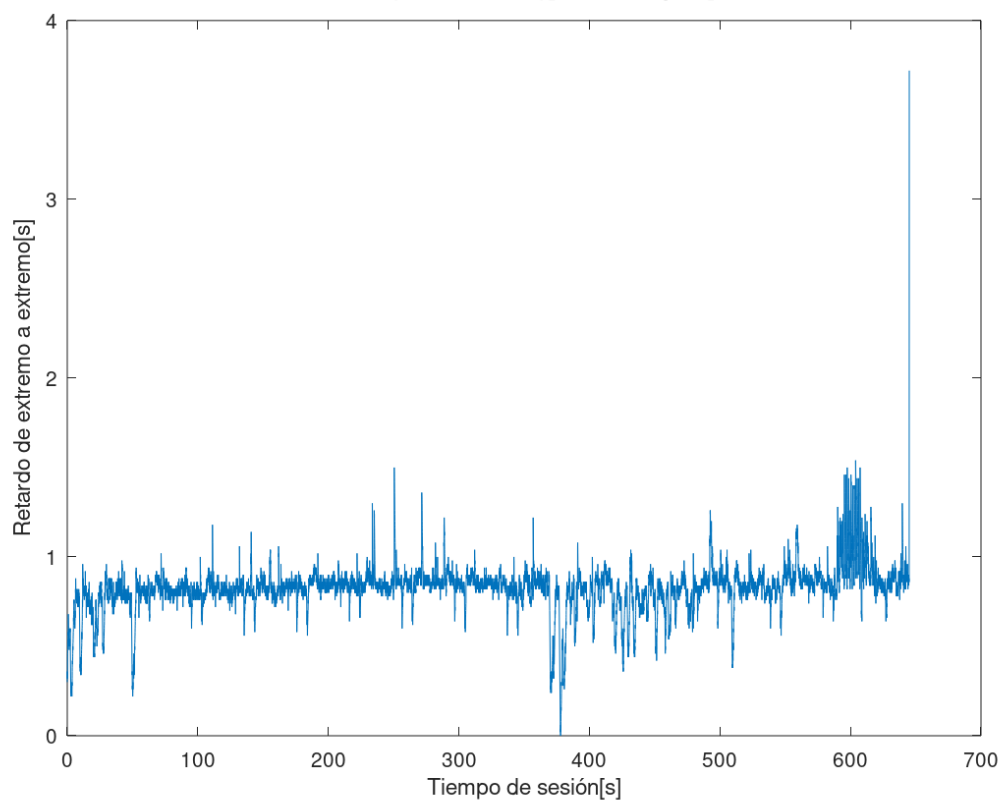
**Traza 4.- Retardo de extremo a extremo
VoIP(Voice over IP)[Macroscópica]**



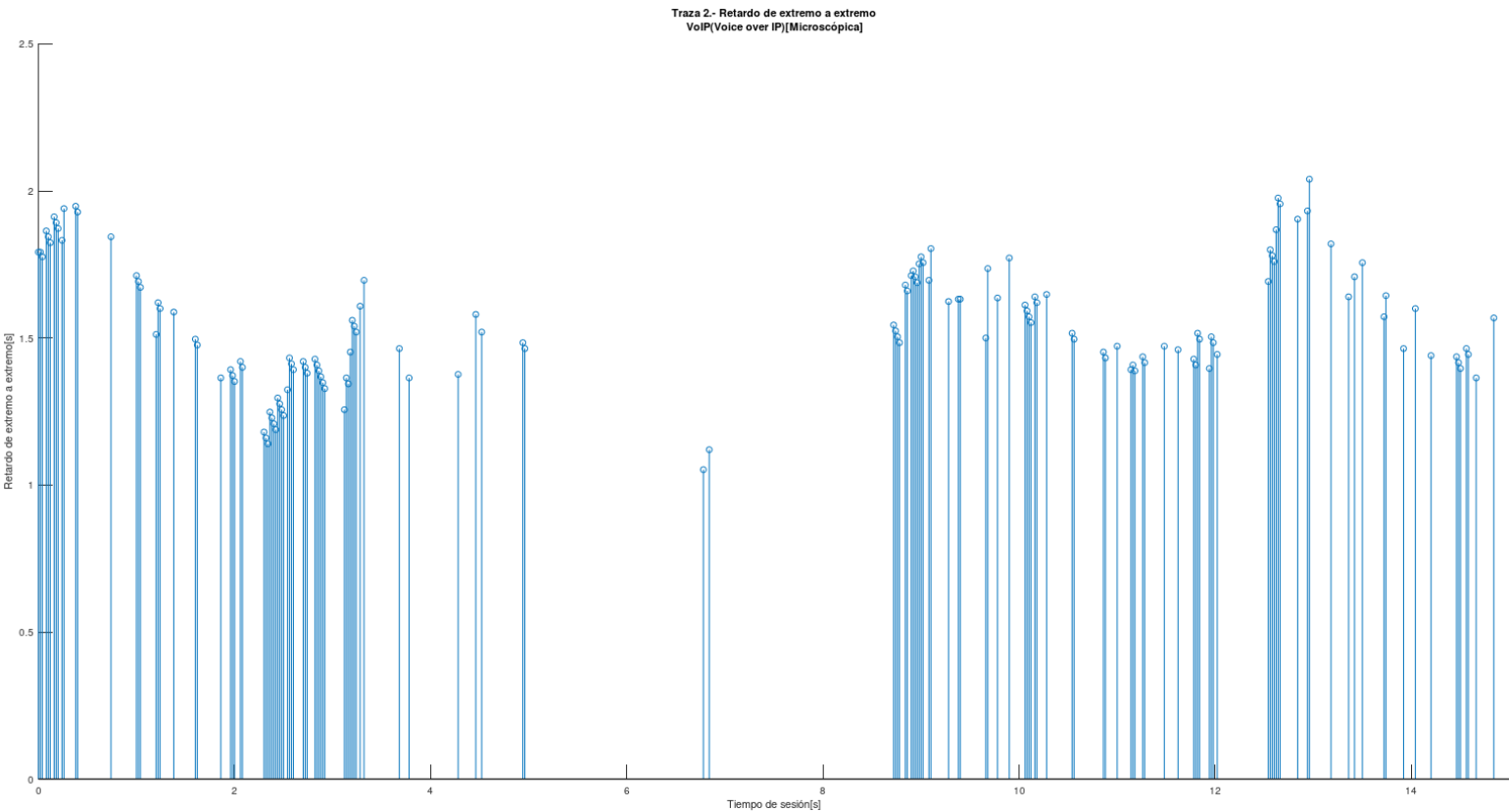
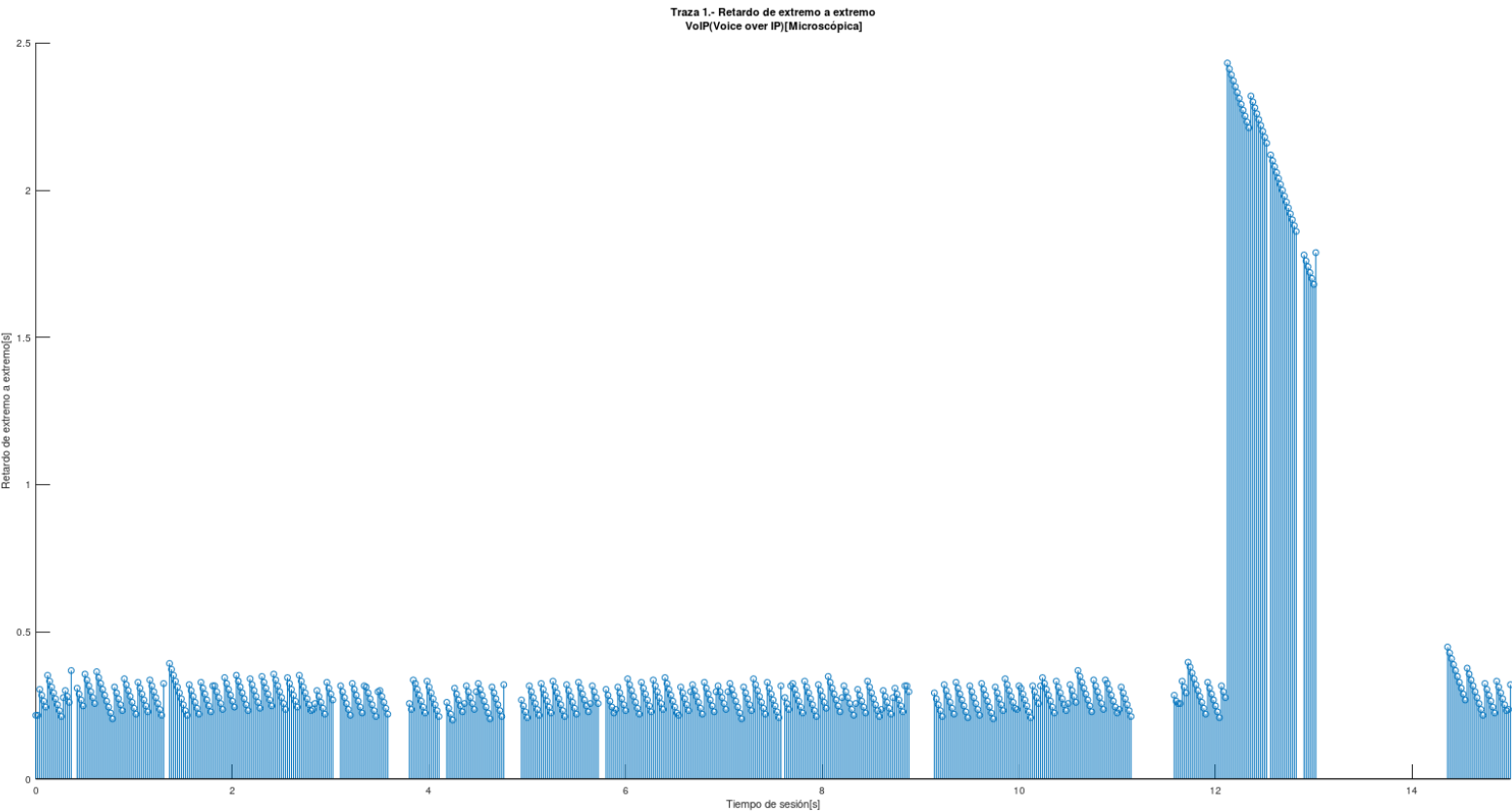
**Traza 5.- Retardo de extremo a extremo
VoIP(Voice over IP)[Macroscópica]**

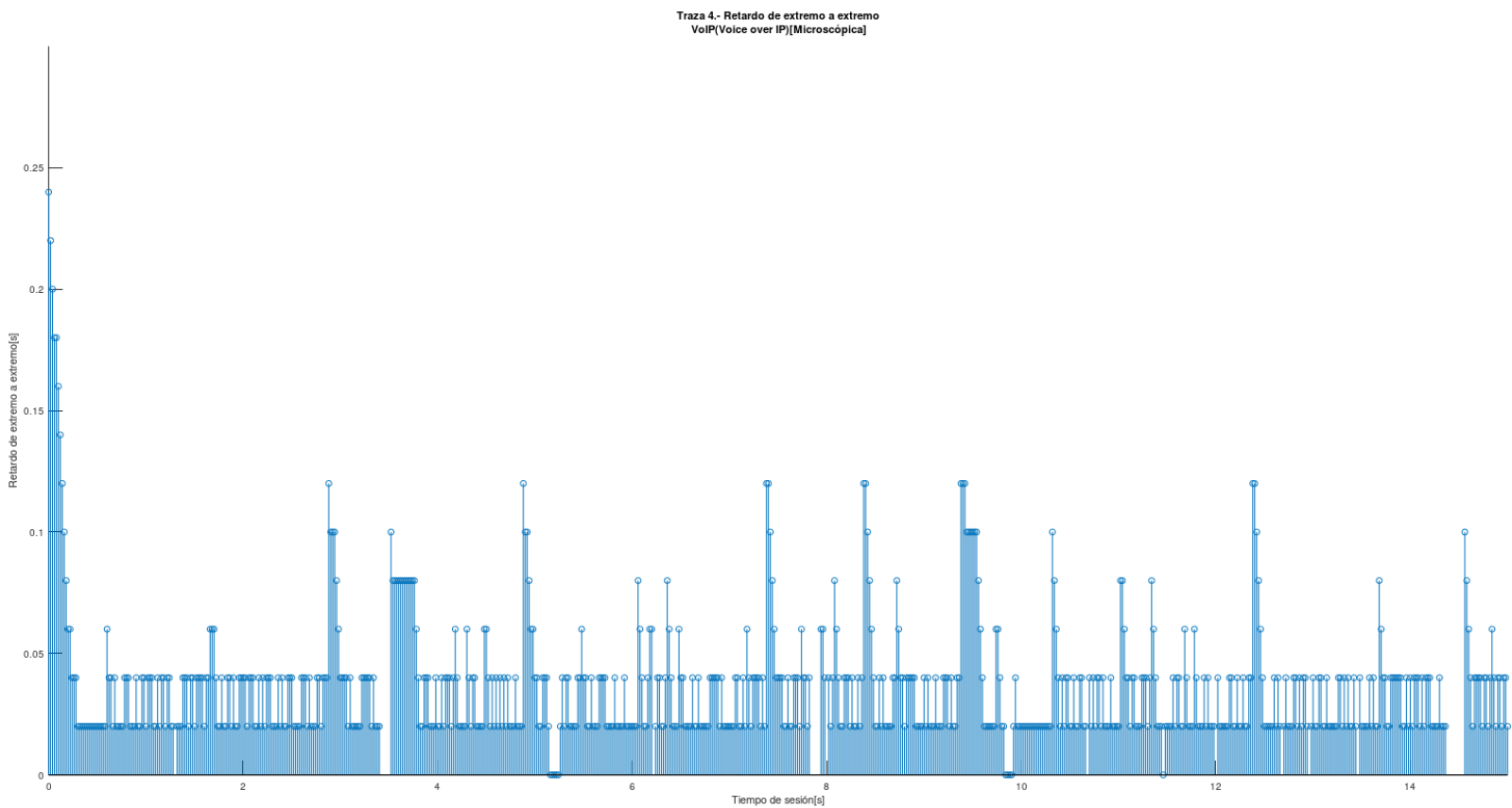
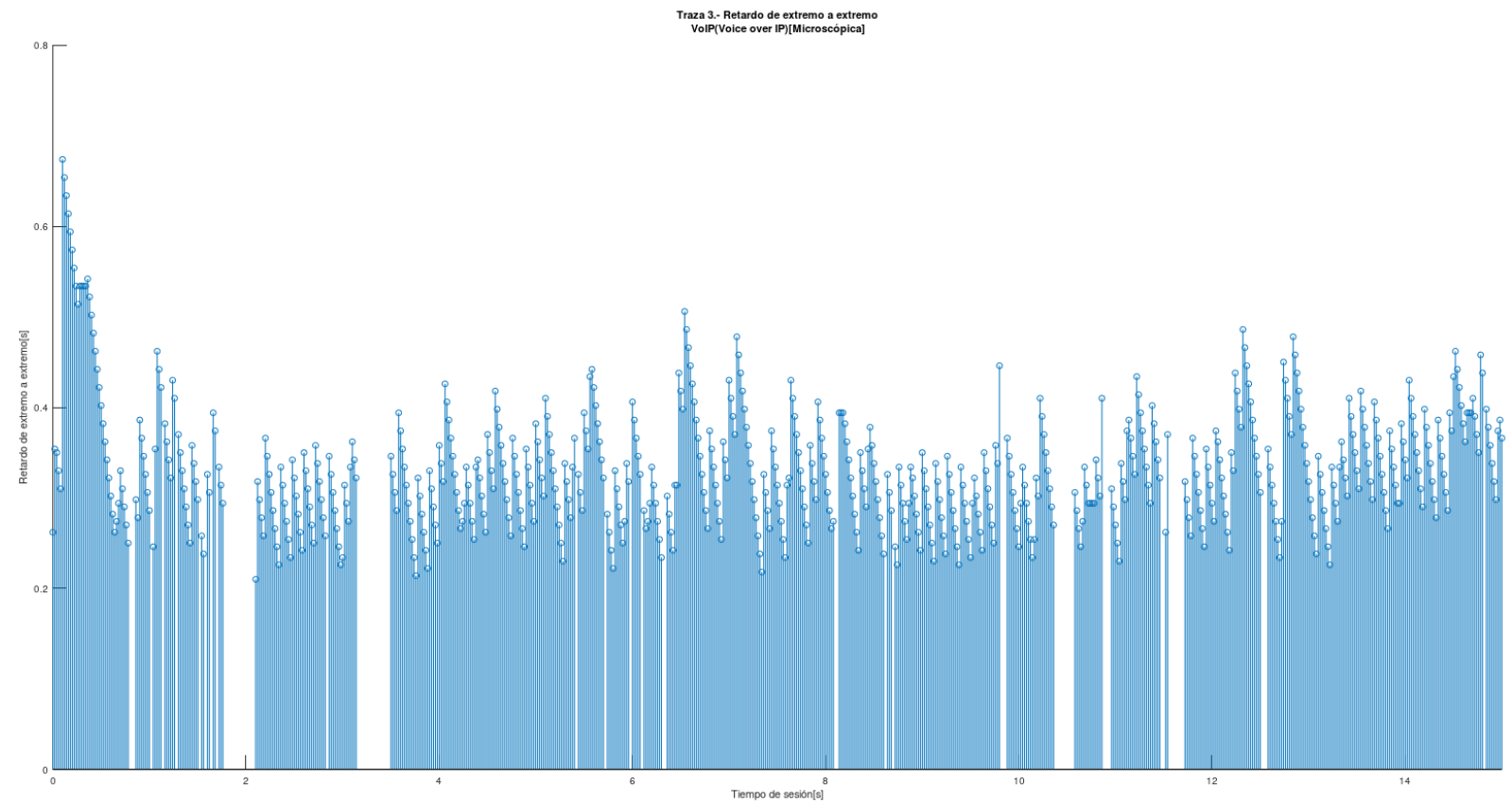


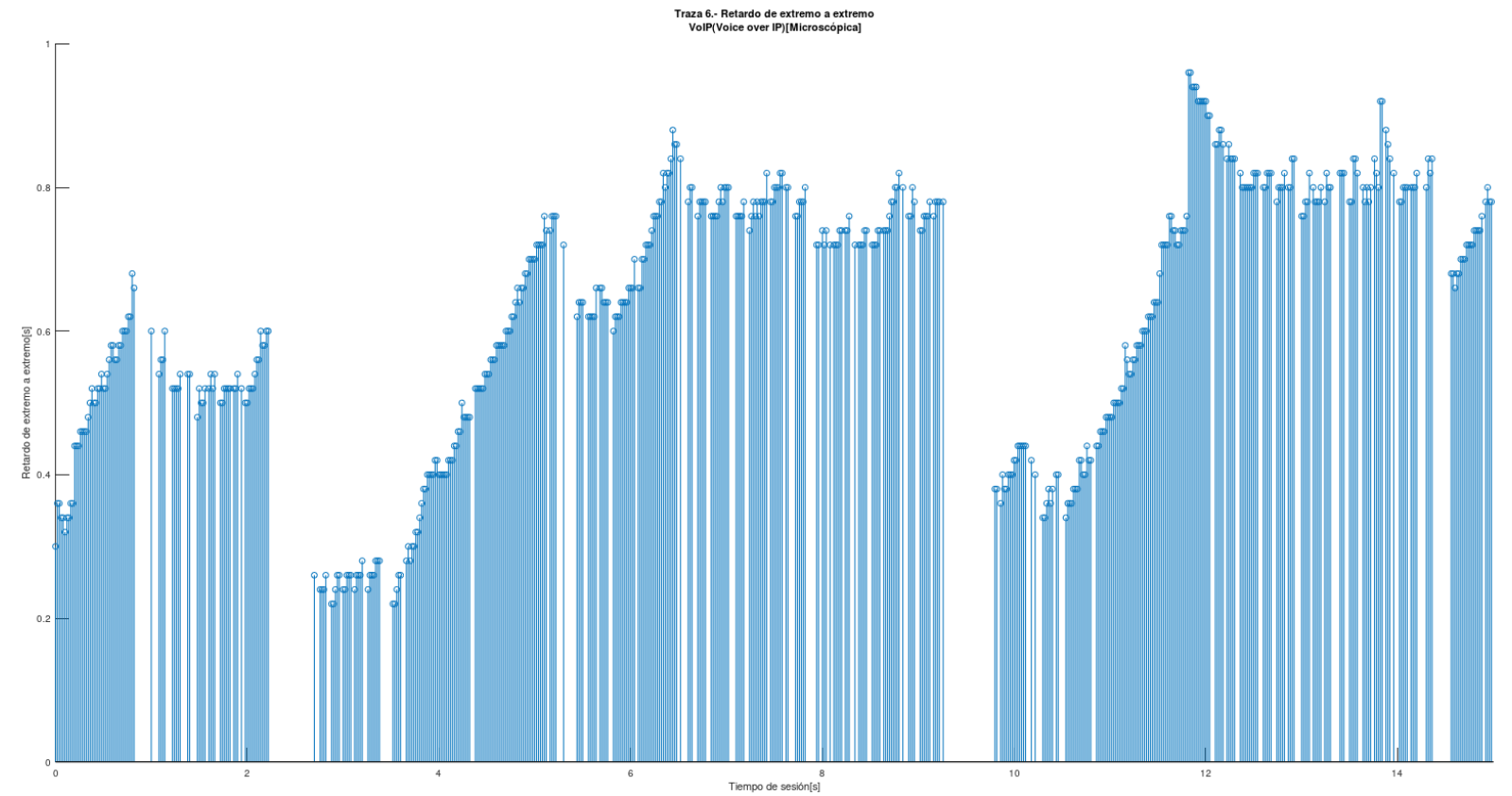
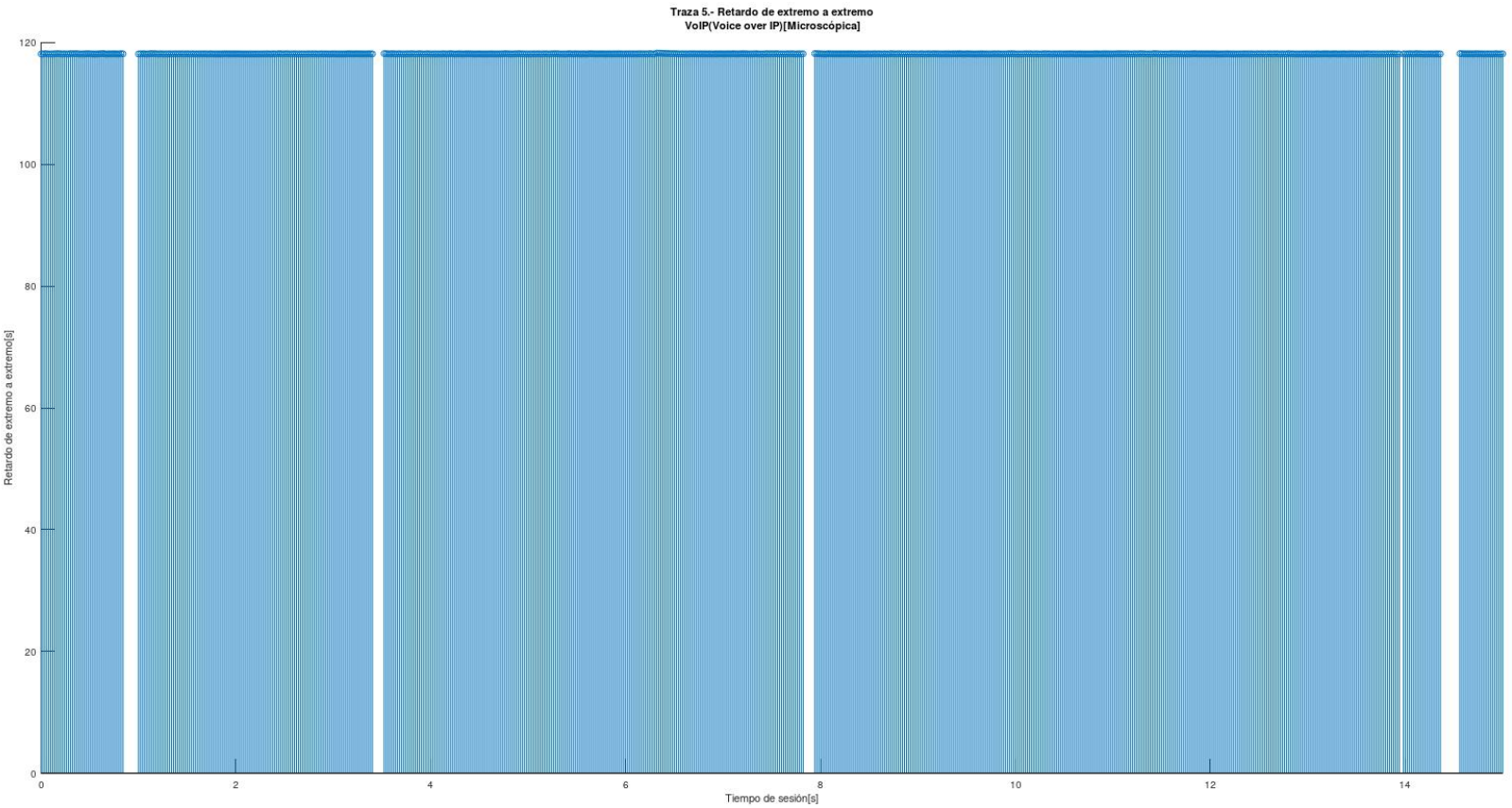
**Traza 6.- Retardo de extremo a extremo
VoIP(Voice over IP)[Macroscópica]**



Gráficas Microscópicas







2. ¿Es posible observar fenómenos particulares en las gráficas que acaba de obtener? Comente al respecto.

En las graficas se puede observar una pendiente que indica que los tiempos del reloj del emisor y el receptor son diferentes, se puede decir que no están sincronizados, es por eso por lo que hay un retraso de la recepción de los paquetes. Y también se puede ver en las gráficas cuando son zonas de silencio.

3. En una tabla, incorpore las estadísticas obtenidas con sus scripts; i.e., número de traza, total de paquetes, total de frases, retardo promedio de extremo a extremo.

NÚMERO DE TRAZA	1ª ESTAMPA DE TIEMPO DE TRANSMISOR	TOTAL DE FRASES	TOTAL DE PAQUETES	MÍNIMA DIFERENCIA	RETARDO PROMEDIO
1	131912079	57796	56979	-641808	675.519
2	119211478	24896	24490	-14643186	760.623
3	135895789	38175	37640	-774206	574.162
4	460510397	28065	27814	-807010	289.542
5	415181337	53375	52836	0	545.416
6	501927541	23591	23293	-1489551	320.781

4. Investigue al menos dos métodos propuestos en la literatura para eliminar la desviación de la pendiente en el retardo. Describa brevemente su principio de funcionamiento.

1.- Uno de los métodos que puede ayudar a disminuir la pendiente del retardo, es la utilización de buffers de playout. Las aplicaciones multimedia en internet utilizan este mecanismo en el receptor para suavizar la variabilidad del retardo de extremo a extremo. Este procedimiento es aplicable para la transmisión de audio como para la transmisión de video. Consiste en que los paquetes que lleguen después de un tiempo determinado (tiempo de reproducción programado) se consideran perdidos y no se reproducen.

2.- En el artículo publicado por Sue Moon hace un análisis de tres algoritmos que implementa la eliminación de la pendiente, dos de ellos nos permite el cálculo de los límites superiores e inferiores del retardo de reproducción, el promedio óptimo para un determinado número de paquetes perdidos de una traza, el tercer algoritmo, el cual es el que propone Sue Moon, nos permite un ajuste flexible del retardo. Con este algoritmo se hace un seguimiento del retardo de la red para los paquetes recibidos y mantiene de forma eficiente la información.